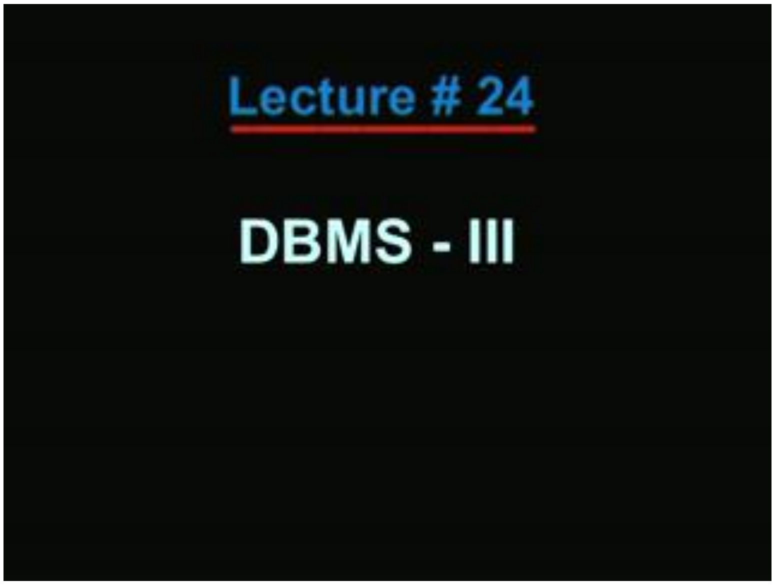
Management Information System
Prof. Biswajit Mahanty
Department of Industrial Engineering & Management
Indian Institute of Technology, Kharagpur

Lecture No. # 24
System Design – II
DBMS- III

(Refer Slide Time: 00:50)

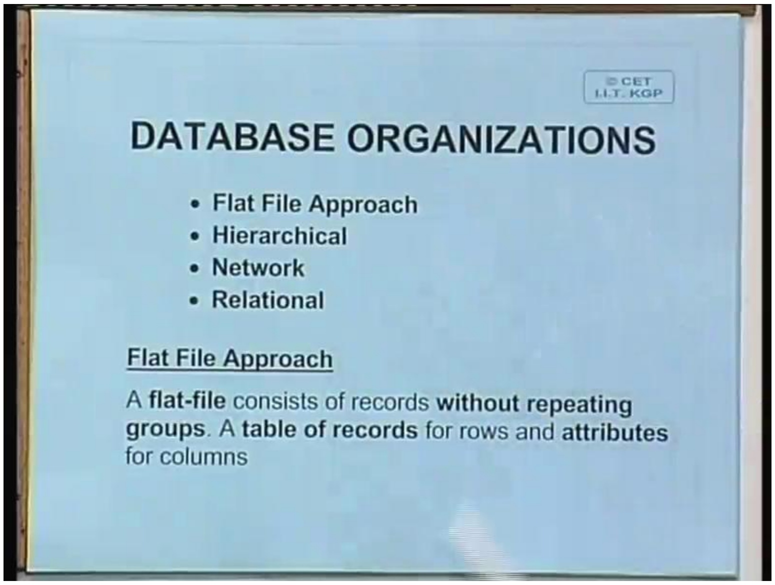


Lecture # 24

DBMS - III

So we are in the middle of the discussion on the database organizations. What I said on the last class that there are 4 types of database organization. The flat-file approach, the hierarchical, network organization and the relational organization.

(Refer Slide Time: 01:11)



© CET
I.I.T. KGP

DATABASE ORGANIZATIONS

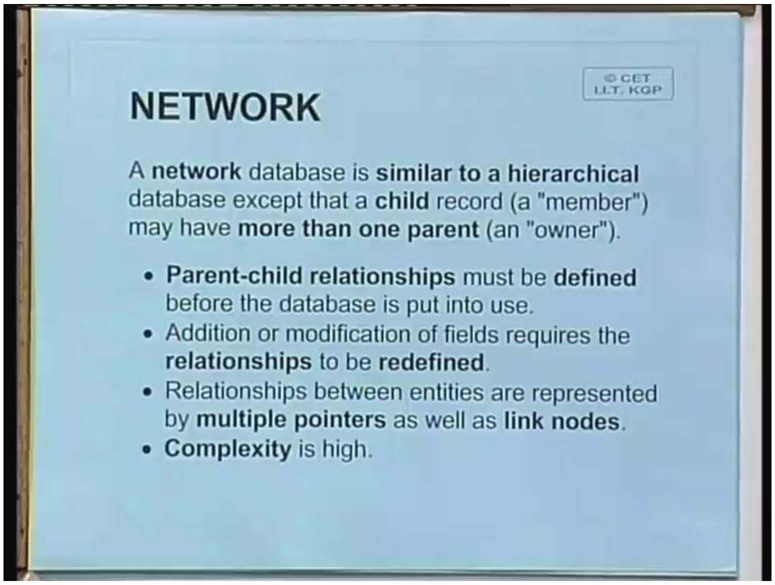
- Flat File Approach
- Hierarchical
- Network
- Relational

Flat File Approach

A **flat-file** consists of records **without repeating groups**. A **table of records** for rows and **attributes** for columns

Now flat file approach is basically does not follow any particular structure. But it is you can assume it is a simple file without any interconnection between files. The only thing, there are no repeating groups fine. Then the hierarchical is a like a tree structure the usual parent child kind of structure and advantage is if the query is from parent to the child all right. Suppose student to the subject, then you have to answer queries like what are the subjects taken by a given student. What are the grades obtained by him, easy to answer. But the reverse 1 for example from subject to student who are the students we have taken a given subject will be difficult to answer.

(Refer Slide Time: 02:04)



© CET
I.I.T. KGP

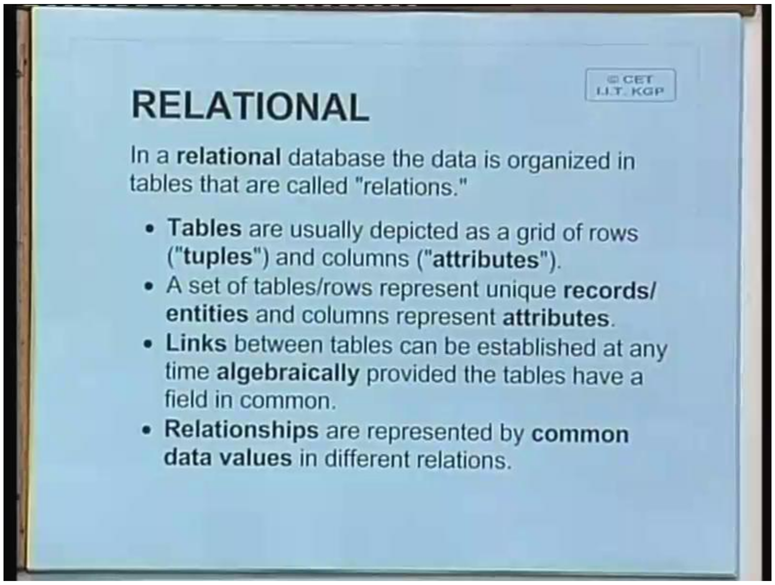
NETWORK

A **network** database is **similar to a hierarchical** database except that a **child** record (a "member") may have **more than one parent** (an "owner").

- **Parent-child relationships** must be **defined** before the database is put into use.
- Addition or modification of fields requires the **relationships** to be **redefined**.
- Relationships between entities are represented by **multiple pointers** as well as **link nodes**.
- **Complexity** is high.

Network on the other side is both ways parent to child as well as child to parent in the sense that 2 sets of records are on the same level. And they are connected through link nodes which can have additional information. And we can if the access paths or network connections are already established then you can answer those kind of queries this we have discussed.

(Refer Slide Time: 02:35)



© CET
I.I.T. KGP

RELATIONAL

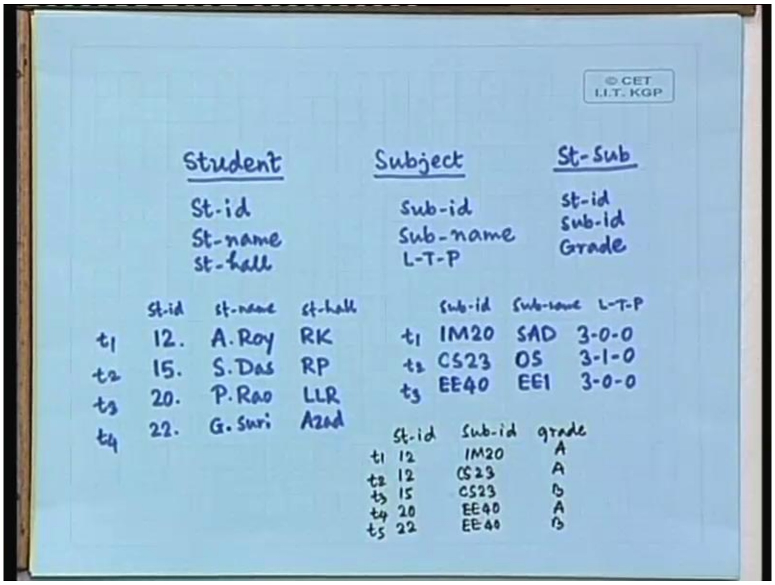
In a **relational** database the data is organized in tables that are called "relations."

- **Tables** are usually depicted as a grid of rows ("**tuples**") and columns ("**attributes**").
- A set of tables/rows represent unique **records/ entities** and columns represent **attributes**.
- **Links** between tables can be established at any time **algebraically** provided the tables have a field in common.
- **Relationships** are represented by **common data values** in different relations.

The relational structure the database organization is a totally different kind of organization right. Here whenever you are adding, let us say a student has taken 3 more subjects all right. In terms of hierarchical or relational structure it means actually putting an access path or a pointer you see setting an access path or a pointer from the student record to the 3 subject record. This is what happens in hierarchical or network structure. So you can see that every add delete modify is a cumbersome process in hierarchical or network structure because it requires a change in the database organization. The relational database organization on the other side purely depends on algebra right a special kind of algebra which is called relational algebra in short RA fine. So here the basic idea of relational structure is that we have tables, we have the tables which are usually depicted as a grid of rows usually called tuples right and columns which are called attributes.

A set of tables or rows represent unique records entities and the columns are representing the attributes links between tables can be established at any time algebraically that is what is known as relational algebra provided the tables have a field in common. You must have at least 1 field in common sometimes more than a field in common so that you can carry out the relational algebra and provide the answer to the queries the relationships are represented by common data values in different relations. We will explain what exactly is meant by that. Suppose you have the relation between student and subject say you have a student file having the student records you have another file called the subject file having the subject record. But how do you know that which student has taken which subject. See what exactly I am trying to say is this that we have a student record.

(Refer Slide Time: 05:34)



© CET
I.I.T. KGP

Student

St-id
St-name
st-hall

Subject

Sub-id
Sub-name
L-T-P

St-Sub

st-id
sub-id
Grade

	St-id	st-name	st-hall
t1	12.	A.Roy	RK
t2	15.	S.Das	RP
t3	20.	P.Rao	LLR
t4	22.	G.Suri	Azad

	Sub-id	Sub-name	L-T-P
t1	IM20	SAD	3-0-0
t2	CS23	OS	3-1-0
t3	EE40	EE1	3-0-0

	St-id	Sub-id	grade
t1	12	IM20	A
t2	12	CS23	A
t3	15	CS23	B
t4	20	EE40	A
t5	22	EE40	B

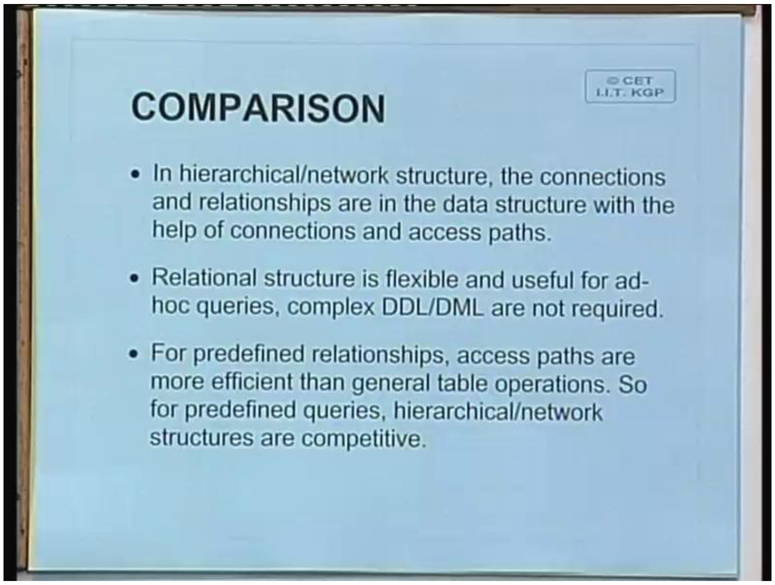
We have a subject record. So from the student record we get certain student roll numbers, names, etcetera. From subject record we have some subject name roll number etcetera. But how are they related which student has taken which subject in hierarchical relational structure you have the say let us take tuples as well. Say we have some students and we have some subject details right. Now the question is see these are like st id. This is st name, this is st hall, this is subject id, this subject name, this is LTP lecture tutorial projects. Now what is a tuple this student record I have taken 3 tuples. See this is the concept of tuple t1, t2, t3 how many tuples in student record 3 I have taken you can also consider fourth tuple say roll number 22 right. So you can I have taken 4 tuples here and 3 subject tuples. Now in network structure what you have to do suppose A Roy has taken all the 3 subjects and you have to define a pointer to each of these 3 subjects and in between you find a link node where you can probably keep the grade. In relational algebra what you need to do you have to establish it by taking a relationship table as well.

So how to get the relationship table usually through by considering the primary keys of both the tables all right. So we have to have a third table, let us call it st sub. See what is in a name you can even call it student takes subject all right. You can call it st sub. I mean it is just a name please remember that student take subject and you know it it is like easy to here. So we may even name the relationship as takes but that takes is nothing but a name. You can as well call it st sub.

So you can have student id and subject id and any field which is a property of both for example grade grade is a property of both okay. So we can now tell me if this is the student, this is subject then how many tuples st sub can have this is a upper limit the upper limit of the number of tuples of st sub would be 12. Why 12? Because assuming all the 4 students have taken all the 3 subjects. Is it clear? So A Roy, SAD, A Ray, OS, A Roy, EE1, S Das, SAD, etcetera. Now usually it does not happen. Usually it does not happen and we may have less number of tuples in st sub is it clear less number of tuples in st sub. So at least some few tuples let me write right. So I have taken a few tuples of only 5 I have taken.

See interesting question will be now that who are the students who have all got A grade in a given subject. Basically what are the names of the students what are the names of the student all right. Suppose you have a query what are the names of the students who have got A grade or who have got A grade in a subject. So you see the answer cannot be given from 1 table you require more than 1 table all right. So whenever you want to do that you do what is known as a join operation you join the 2 tables right. Say for example who are the students of a particular hostel say RK who have got A grade. You see I require to give the answer from 2 tables. Sometimes you may have to join all the 3 tables right. So there are 3 fundamental processes we will discuss them later selection projection and join usually and many some others as well. So using these 3 fundamental processes selection projection and join usually the queries can be answered all right. So we will take up SQL later on let us before we come to that. Let us see some comparison.

(Refer Slide Time: 14:56)



COMPARISON

© CET I.I.T. KGP

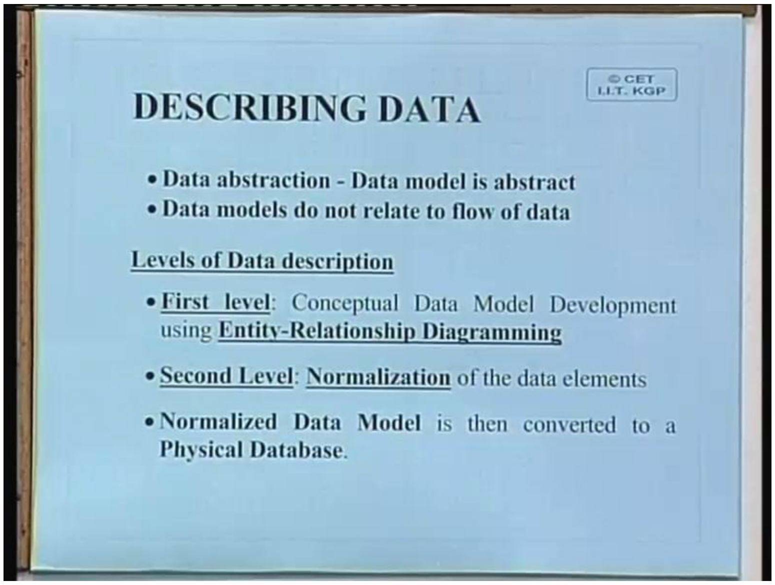
- In hierarchical/network structure, the connections and relationships are in the data structure with the help of connections and access paths.
- Relational structure is flexible and useful for ad-hoc queries, complex DDL/DML are not required.
- For predefined relationships, access paths are more efficient than general table operations. So for predefined queries, hierarchical/network structures are competitive.

This is a comparison of hierarchical or network structure. I have probably have said all this before end in hierarchical or network structure the connection and relationships are in the data structure with the help of connection and access paths all right. The relational structure is flexible and useful for ad hoc queries complex DDL or DML are not required we need not set up the access paths for predefined relationships access paths are more efficient than general table operations. So these are known as the relational algebraic operation known as general table operations I said at least 3 processes projection selection and join and so for predefined queries hierarchical network structures are competitive means as good as relational may be even better right. But unfortunately the world has moved on see what has happened in today's world we cannot depend on what queries will be asked right.

The time has come when you have to make an open kind of a system where any kind of queries can be answered right. We should be prepared because as the businesses are expanding and as data processing or information systems have become more and more complex and see you have to also understand that efficiencies can no more be generated from a single system. A single system the days of efficiently running a single system is over increasingly we have to have combined systems. For example not optimize production alone optimize production and marketing together right. So say that is why we today increasingly here things like enterprise resource planning ERP supply chain management SEM you know including IT, the IT advantage of doing business in a new way globalization of business. We keep on hearing these newer terms which basically draws the total quality management basically draws from integrating several applications.

So when you are integrating several applications today's queries are also going to be across platform, across platform if the marketing decides not to sell too much this week or this period what will be the impact of that in production decisions. So you see this is not a straight forward query which you can give from a single table or a single set of data. You require multiple data to be put together and the queries to be handled in that manner it cannot be called predefined more or ad hoc in nature and 1 should be prepared to answer them. Because of this the relational databases have become all that popular right. So that is about the database organizations. Now let us move on to something more fundamental that is entity relationship diagramming.

(Refer Slide Time: 18:36)



© CET
I.I.T. KGP

DESCRIBING DATA

- **Data abstraction - Data model is abstract**
- **Data models do not relate to flow of data**

Levels of Data description

- **First level**: Conceptual Data Model Development using **Entity-Relationship Diagramming**
- **Second Level**: **Normalization** of the data elements
- **Normalized Data Model** is then converted to a **Physical Database**.

The entity relationship diagramming basically is a part of data description right. So 1 thing about the data description as a very fundamental point you must understand right in the beginning that the data models are abstract reason kind of data abstraction. See it has basically 2 meanings. The meaning of abstraction is usually identifying the relevant portions of an object or an entity right choosing from several others may be attributes or properties and using it for a specific purpose.

This is the basically meaning of abstraction but on a different level the meaning of abstraction is the data description has little to do with the way a particular process is being carried out. See I have given an example from bank you might have seen many banks could be the state bank at IIT Kharagpur or may be other foreign banks somewhere in a city the operations of the banking will be entirely different right.

You may find more of manual processes here and more of automation in a foreign bank right. The kind of things that we were discussing like issues and recedes the issues and recedes probably will be completely automated in an ATM counter. So the data flows or in particular the data flow diagrams could be organizations specific please understand it carefully depending on which organization you are modeling your DFD could be different is it all right. If you model a banking operation in here in Kharagpur or you model it in another city or you model in a foreign country all your data flow diagrams will be different. On the other hand for each of this banking operation if the fields have the same meaning the fields has different meaning that is the different issue. But if the fields have the same meaning and you have the same number of fields the data base design should be exactly same. If you have more fields to consider that’s a separate issue. But if there are same fields however differently they may be organized the operations right. The operations may be very differently organized.

But if it is the same field you have a customer you have the customer name you have the bank account. You have a transaction you have an issue you have a receipts you have a ledger the database design will be exactly same is it clear. That is why the data modeling compared to the data flow modeling is an abstract design. It is an abstract design that is why it is called conceptual data modeling. It is a conceptual diagram it has nothing to do with reality as long as the fields are obtained you get the answer you need not understand how the fields are organized is it okay. So in a way it is easy also in in a different sense why it is easy you simply have to extract the fields and understand their meaning. You do not have to understand the context in which the people are using whether it is a manual ledger or whether it is in floppy diskettes or who is using it some people may decide to put it in 10 ledgers. See in an highly inefficient system highly inefficient system the same information may be in 10 ledgers right.

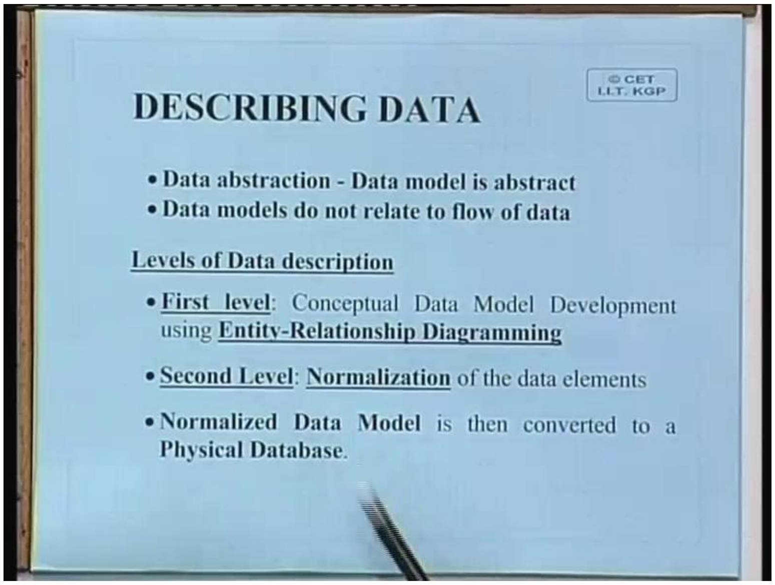
I have seen with my eyes that suppose there is a government organization and by mistake you have written a later to them you have nothing to do with it. You just give them a letter that I do not know why you are treating is like this. I had some business with you but you have not treated me well just give this letter. What will happen let me tell you this data this letter will be first received in the receipt section, there is a receipt section the receipts section there is an incoming ledger. The incoming ledger immediately there will be an entry such and such letter has been received from such and such person see is an elaborate design who has send the letter on which date through which post office it has come so your letter is recorded. Now after your letter is recorded naturally the question is since your letter is vague to whom should it go right? It is that requires certain decision naturally the issue and dispatch clerk cannot make this decision.

So they will send the letter to another department probably a little higher department control department to decide where this letter should be send. Before the letter is being sent there is an outgoing register the outgoing register. You again your letter will be entered there and some clerk will come from the control section peon. The peon will make an entry there and again it will go to the control section the control section again as an input register and an output register all right. Then they will decide okay let us send this letter to Mister X of some department. It will not go directly from the control department to Mister X it will come back to the dispatch section. Obviously with too more entries go to Mister X in this department 1 entry in the incoming register then probably Mister X will write I have nothing to do with it. You will not be surprised if the letter comes back to you we could not understand. What is to be done about your request?

That is okay. That is the end of it. But with nearly 64 entries in different registers of that company, it is reality I am not exactly this happens exactly this happens. So it is an highly inefficient way of managing data highly inefficient way of managing data all right but what I am basically trying to say even if you go to this industry and collect the basic information what is the basic information a letter has come is it all right. A letter has come so what is the information contained in the letter one is who has send it on which date it has come. What is the post office through which it has been posted to whom it has been address to and probably any reference of this letter whether he is a real customer or what is he and may be later on the content of the letter is it clear. So even if it is an highly inefficient company example I gave or highly efficient company right where the reception who ever receive the clerk moment. They see the letter make a phone call that such a letter has come please tell me what should I do with it. That is all. That is the end of it the decision is taken then and there and an answer is scripted to you all right.

Even then there if they want to keep an record of this letter the database content to be exactly the same all you need to do find the data fields do the relational analysis through entity relationship diagramming through normalization and come up with the same conceptual data model all right. It is independent of the data process which is available there. So there are data models do not relate to the flow of data that is the key point. In fact probably this is the reason why the object oriented analysis and design methods have become so popular. You see since the data modeling or the database design is an abstract way of doing things and little to do with data processes. It has got lot of advantages the advantages is that it can be reused right it is not dependent on the present situation. So this is why whenever object oriented analysis is done essentially it has you can say it is an offshoot of the year diagram an offshoot of database design all right. But only thing you do not call them entities call them objects and not only attributes but you also associate methods and definitely much more than this definitely much more than this but we shall come back to it in our next set of classes.

(Refer Slide Time: 28:46)



© CET
I.I.T. KGP

DESCRIBING DATA

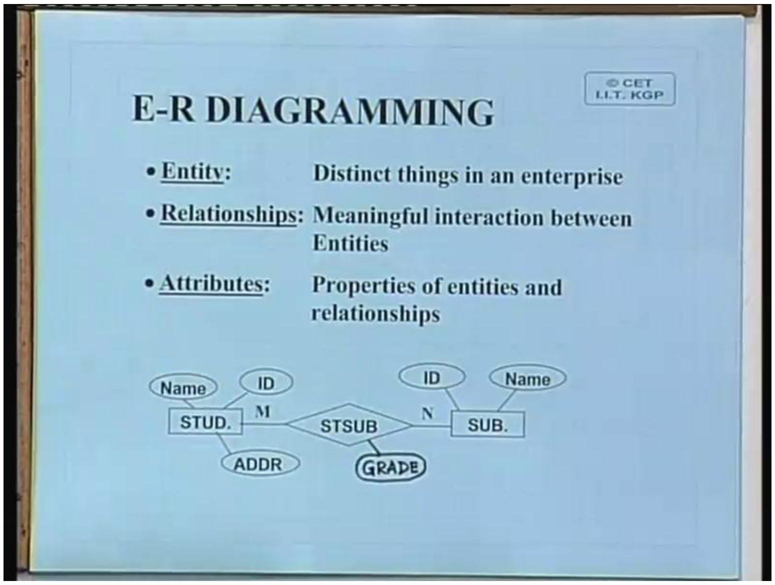
- **Data abstraction - Data model is abstract**
- **Data models do not relate to flow of data**

Levels of Data description

- **First level**: Conceptual Data Model Development using **Entity-Relationship Diagramming**
- **Second Level**: **Normalization** of the data elements
- **Normalized Data Model** is then converted to a **Physical Database**.

Now when we are describing data there is essentially 2 levels. As I was saying in the first level we make a conceptual data model by using entity relationship diagramming and the second level we carry out the normalization of the data elements. The normalized data model is then converted to a physical database so that is the basic process of data description. Let us took at what is an entity relationship diagram.

(Refer Slide Time: 29:13)



© CET
I.I.T. KGP

E-R DIAGRAMMING

- **Entity:** Distinct things in an enterprise
- **Relationships:** Meaningful interaction between Entities
- **Attributes:** Properties of entities and relationships

Name
ID
STUD.
ADDR
M
STSUB
GRADE
N
ID
Name
SUB.

The entity relationship diagram essentially has got 3 components. As the name suggests the entity look at in this small diagram the student and the subject these are the 2 entities all right. So between these 2 entities we have certain attributes so definition of entity distinct things in an enterprise about which data is being collected. See if you are not going to collect data about it probably is better not to call it an entity a small example students live in a hostel or a hall. Now hall is an entity or not yes or no. Why it depends? It is an entity. Why not look at this hostel is a distinct thing in an enterprise yes or no definitely all right. Now the question is whether we are interested in collecting data about it. Suppose we are trying to also collect hall why is information then we should not simply say you know the hostel the student is staying.

We just do not put in attribute we also say student hostel right. Hostel and between them we may have a relationship fine and therefore student will have only hall number and the hall will have its own details hall number hall name etcetera, etcetera. So entity relationship the relationship is the meaningful interaction between entities between the student and the subject we have an entity. I mean not an entity a relationship we may call it takes student takes subject and I have it is not shown here but you can always show an entity called grade. Now tell me suppose we do not want to include grade if we do not want to include grade is the relationship required. Understand the question. Suppose we do not want to keep grade in our database right. So there is no grade. Grade is not given yet. So do you still want the relationship table to be included?

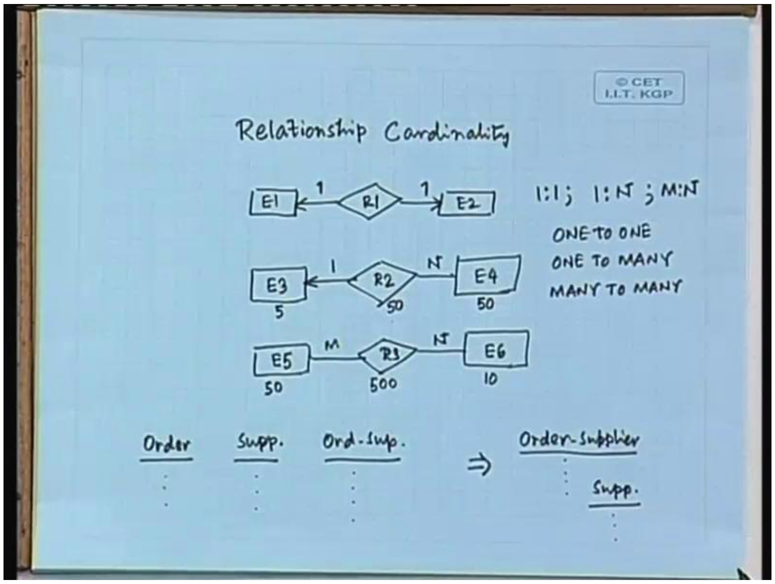
Student: Yes sir, yes sir.

It is required the reason being which student has taken which subject these questions can be answered. Now attributes. Attributes are properties of entities and or relationship. What is the difference between property and attributes and entity? See the student is an entity this is the distinct thing anything that describes it is an attribute. So do not confuse between attributes and entities please remember the whenever later on we discuss about things like normalization and other kind of things we must very, we must be very, very clear about an entity and its attributes these are 2 different things. Otherwise very common mistakes may come in. What is this M and N? See I have written an M I have written an N here. What are these?

Student: Tuples.

Not tuples these are called relationship cardinality relationship cardinality. Let me write about this.

(Refer Slide Time: 35:02)



Relationship Cardinality

1:1 ; 1:N ; M:N

ONE TO ONE
ONE TO MANY
MANY TO MANY

So some examples there are 3 kinds of relationship cardinality: one is to one, one is to N and M is to N. One is to sorry one is to 1 one is to N and M is to N. So 1 to 1 they are called 1 to 1, 1 to many and many to many right. One to one, one to many and many to many say a many to many relationship such as student to subject suppose if E 5 is student E 6 is subject and ER 3 is st sub st sub. So what does it mean 1 student can take many subjects fine true or false true. One subject can be taken by many students clear. So this is why it is a many to many relationship okay. Think of the same example where there is only 1 subject it could be peculiar. But suppose there is only 1 subject and there are many students right like in you know some ancient Indian universities not really university some very famous guru is teaching a single subject many students are coming to him and learning that subject.

So in this particular in the context of this very limited scenario we have 1 subject I mean look at this way this is student subject if it is there. So one student can take, I am sorry. This is subject, this is student. So the subject is given too many students but the student can take only 1 subject. So that is called 1 to many relationship is it clear 1 to many relationship and in 1 to many relationship we usually draw an arrow on the 1 side. See these arrow does not mean it is flowing towards 1 or any such thing this arrow has nothing to do with flow right. So it just a matter of indication to identify that this is the side of 1. Is it clear? These arrow 1 arrow you can see these arrow simply denotes that this is the 1 side of the cardinality relationship cardinality 1 to many clear. Now 1 to 1 relationship very, very difficult to obtain because in real life a hardly anything that is 1 to 1 and if it is so much true to 1 to 1 then probably you can merge them together. There is really no need to put it into so much elaborate structure you can as well define them in 1 entity itself all right very, very difficult.

Suppose in an institute you have just 1 student and there is only 1 subject and this situation will never change situation will never change. Then you can say it is an 1 to 1 relationship right. Suppose you have separate records for a particular cheer professor you know there are some foreign universities they call some chair professors. So the chair professor and the professor, right. Only 1 professor can become that chair professor and in that chair professor at a time only 1 professor can be there this is a kind of 1 to 1 relationship. So 1 to 1 relationships are basically no point in discussing much let us say theoretical possibility in all practical situations 1 to 1 is almost ruled out does not really there. A very peculiar thing about 1 to many relationship not always but many time whenever you have an 1 to many relationship. First of all let us see many to many. See in a many to many relationship I have just I was trying to show you.

Suppose there are 50 students and there are 10 subjects. So theoretically how many combinations are possible 500. So you can see that in a many to many relationship the relationship tables are usually larger than both the, from both the sides is it okay. But a peculiar thing about 1 to many say an order and supplier. How a supplier how many orders can be given 1 or more a given supplier over period can have many orders yes or no right but can a supplier can a given order be given to more than 1 supplier sometimes yes in peculiar cases yes sometimes an order is shared between 2 suppliers is a very peculiar cases are there. But in most 99 percent of the situations and 99 percent of the organizations, it is not the situation. For example IIT Kharagpur more or less gives orders to single suppliers. So a supplier can have many orders whereas an order can have only 1 supplier so which is order which is supplier you tell me this is order or this is supplier. This is supplier.

This should be supplier why a, see a supplier can have many orders and order can have only 1 supplier fine. So these are some examples of 1 to many. But let us say there are say 5 suppliers there are 5 suppliers okay and there are 50 orders. So how many will be there in R 2 it will be 50. Anybody having any doubt? You try to understand, there are 5 suppliers all right and there are 50 orders all right. So these 50 orders, now if you try to combine them but the combination will be always 1 to 1 because a given order will be associated with a single supplier only is it not a given order. Cannot be associated with all 5 if a given order can be associated with all 5 you would have had 250. But since a given order has a single supplier there will be exactly 50 in the relationship. Now a very peculiar thing has happened you have 50 here you have 50 here. So what is possible and 1 to many relationship. Suppose we have an order file we have a supplier file and we have an order supplier file.

We can replace it by only the order supplier file and the supplier file we need not have a separate order file can you see this? Please understand it very carefully. It is an 1 to many relationship since the number of records here that is 50 number of records, 50 here it is exactly going to be the same all right. So instead of keeping 3 records 3 separate tables we can keep a supplier table. We can keep an order supplier table or in other words in the order file itself we can add the supplier number instead of separately keeping order and supplier in a relationship table. See we are calling it order supplier. But it is nothing but the order table only. So we have this supplier table and in the order table we also keep the supplier information. So in a sense what was being done by 3 tables could be possible to do in 2 tables. But it is not a rule exceptions do exist exceptions do exist. There are situations where these generalization may not be possible but you can say please understand more number of tables you are defining more joints you have to do and more joints means less will be the response time. The responses to the queries are the database processing time will be that much higher. So one need to keep this in mind and do things accordingly okay. So let us close it here because anyway we have been delayed today. Continue from here next class.